2

AD-A236 139

rnegie Mellon University
oftware Engineering Institute

The Software Technical Review Process

Curriculum Module SEI-CM-3-1.5

91-00857

91 5 30 020

The Software Technical Review Process

SEI Curriculum Module SEI-CM-3-1.5

June 1988

Accession For
NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification

By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

James S. Collofello
Arizona State University

QUALITY INSPECTED 1

Carnegie Mellon University
Software Engineering Institute

This work was sponsored by the U.S. Department of Defense.
Approved for public release. Distribution unlimited.

This technical report was prepared for the

SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

JOHN S. HERMAN, Capt, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1988 by Carnegie Mellon University.

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Service. For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

The Software Technical Review Process

Acknowledgements

I would like to thank Jane Siegel and Priscilla Fowler for their valuable and timely input for this module.

Contents

A support materials package, SEI-SM-3, is available for this module.

The Software Technical Review Process

Module Revision History

Version 1.5 (June 1988)	corrections to outline numbers
Version 1.4 (May 1988)	minor typographical corrections
Version 1.3 (April 1988)	format changes, minor corrections, and reference added
Version 1.2 (July 1987)	format changes for title page and front matter
Version 1.1 (April 1987)	slight cosmetic changes
Version 1.0 (September 1986)	original version

The Software Technical Review Process

Capsule Description

This module consists of a comprehensive examination of the technical review process in the software development and maintenance life cycle. Formal review methodologies are analyzed in detail from the perspective of the review participants, project management and software quality assurance. Sample review agendas are also presented for common types of reviews. The objective of the module is to provide the student with the information necessary to plan and execute highly efficient and cost effective technical reviews.

Philosophy

This module provides the depth required to fully understand the software technical review process. This process is a complex group activity for which there exists an abundance of basic concepts evolved over years of practical experience. This module collects these concepts and years of practical experience and integrates them from the software developer, manager and quality assurance perspectives.

Many of the basic concepts in this module are applicable to the introductory course on Software Engineering and, thus, can be used to support the course. In particular, the rationale for software technical reviews, the types of technical reviews which may occur on a project and an overview of review methodologies are important topics for this course.

This module can also be used to support any other modules such as Software Requirements Specification Overview and Introduction to Design which produce documentation that needs to be reviewed. Sample checklists and review agendas are presented for many typical project technical reviews. This module also is important for those modules which describe planning activities necessary on a project, including Software Development Plans and Software Quality Assurance Plans.

This module is also essential to the development of any Software Testing or Software Quality Assurance "course" in which much of the contents of this module can be incorporated or assumed to be prerequisite material.

Objectives

A student who has worked through this module is expected to:

- be able to explain the rationale for software technical reviews;
- understand the broad spectrum of review processes and their range from very formal to very informal;
- describe the various types of reviews that may take place on a project;
- discuss the role of internal and external standards in respect to technical reviews;
- understand the process for assessing the effectiveness of a particular type of technical review in a development process;
- be cognizant of the various types of review methodologies which exist and how to select the appropriate methodology for a particular review;
- be able to effectively execute the role of review leader, recorder, reviewer and producer;
- understand techniques for conflict resolution;
- understand the planning steps for effective reviews including how to select participants, when to schedule a review and how much time to allocate to the review process;

- be capable of developing agendas for various types of reviews;
- describe the content of review reports, the various perspectives for these reports, the possible distribution of these reports and the behavioral impact of report distribution for the producer;
- be able to generate action items as a consequence of a technical review and describe approaches for following up on action items;
- appreciate the complex behavioral factors involved in review processes and be able to utilize techniques designed to motivate reviewers and reduce stress often associated with reviews;
- be capable of interpreting data generated from review processes in terms of both the software undergoing review and the development process and/or maintenance process that produced it.

Prerequisite Knowledge

This module assumes an understanding of the software development and maintenance process at a level where these life cycle activities can be understood in terms of their reviewable work products. The material can, thus, be integrated into all relevant courses including the Introductory Course on Software Engineering after the overall life cycle is discussed.

Module Content

Introduction

The terms *review process* and *software* are used with precise meanings in this section.

A *review process* can be defined as a critical evaluation of an object. Although the term review process often has many connotations, particularly for those with industry experience, the intent of this module is to use this term in its most general sense. Thus, walkthroughs, inspections and audits can be viewed as forms of review processes.

The term *software* is used in this module to apply to all of the work products generated during the development and maintenance of the product and not just the code. The intent of this module is to describe review processes applicable to all of these work products.

Outline

1. Review Processes in Society
    - a. Typical review processes in society
    - b. Roles of review participants for typical reviews
    - c. Training and preparation of participants for typical reviews
    - d. Reasons why some reviews are stressful
    - e. Stress manifestation in review processes
    - f. Basic techniques for minimizing stress
    - g. Techniques for conflict resolution
    - h. Factors essential for successful reviews
2. Rationale for Software Technical Reviews
    - a. Error prone software development and maintenance process
    - b. Inability to test all software
    - c. Reviews are a form of testing
    - d. Reviews are a way of tracking a project
    - e. Reviews provide feedback
    - f. Educational aspects of reviews
    - g. Impact on morale
    - h. Impact on maintainability
3. Types of Software Technical Reviews
    - a. Development life cycle models
    - b. Current standards
4. Behavioral Factors
    - a. Motivating reviewers
    - b. Small group theory
    - c. Minimizing stress
    - d. Review logistics
5. Formal versus Informal Reviews
    - a. Terminology
    - b. Informal reviews
    - c. Need for formalism
    - d. External/internal reviews
    - e. Responsibility of reviewers
    - f. Review reporting and follow-up
6. Review Methodologies
    - a. Walkthroughs
    - b. Inspections
    - c. Audits
7. Roles of Review Participants
    - a. Review leader
    - b. Recorder
    - c. Reviewer
    - d. Producer
8. Planning for the Review Process
    - a. Selecting participants
    - b. Scheduling the review
    - c. Developing review agendas
9. Review Reports
    - a. Management perspective
    - b. Customer perspective
    - c. Developer perspective
    - d. SQA perspective
10. Assessing the effectiveness of technical reviews
    - a. Error detection efficiency
    - b. Cost effectiveness
    - c. Relationship of reliability assurance techniques
    - d. Tool support for review processes

Annotated Outline

1. Review Processes in Society

This section establishes a framework for viewing software technical reviews from a larger scope. The intent of this section is for the students to realize that review processes take place all of the time in our society. Examples of typical reviews include course final exams, job interviews, performance evaluations and IRS audits. Examples of reviews that involve evaluation of team efforts should also be presented. In each type of review it is possible to identify various roles that the review participants perform. For each of these roles varying degrees of training and preparation can be observed depending upon the type of review.

This section also addresses the notion that many reviews are stressful and, therefore, viewed negatively by the review participants. Sources of stress in review processes, observations on how this stress affects the review process as well as basic techniques for minimizing stress in reviews should be discussed.

Conflicts are also a real part of many review processes. Sources of conflicts in reviews should be discussed as well as practical techniques for dealing with these conflicts.

The section concludes with a recognition of the factors essential for any type of review to be successful. These include careful timing of the review, credibility of the reviewers and integrity of the review process.

a. Typical review processes in society

b. Roles of review participants for typical reviews

c. Training and preparation of participants for typical reviews

d. Reasons why some reviews are stressful

e. Stress manifestation in review processes

f. Basic techniques for minimizing stress

g. Techniques for conflict resolution

h. Factors essential for successful reviews

2. Rationale for Software Technical Reviews

This section describes the importance of technical reviews in the software development and maintenance life cycle. The intent of this section is for the students to realize that review processes are absolutely essential to the development and maintenance of quality software [Weinberg84].

a. Error prone software development and maintenance process

The complexity and error-prone nature of developing and maintaining software should be demonstrated with statistics depicting error frequencies for intermediate software products. These statistics must also convey the message that errors occur throughout the development process and that the later these errors are detected, the higher the cost for their repair [Boehm76].

i. Complexity of software development and maintenance processes

ii. Error frequencies for software work products

iii. Error distribution throughout development phases

iv. Increasing costs for error removal throughout the life cycle

b. Inability to test all software

This section must convince students that it is not possible to test all software. Clearly exhaustive testing of code is impractical. Current technology also does not exist for testing a specification or high level design [McKissick84]. The idea of *testing* a software test plan is also bewildering. Testing also does not address quality issues or adherence to standards which are possible with review processes.

i. Exhaustive software testing is impossible

ii. Intermediate software products are largely untestable

c. Reviews are a form of testing

The section fosters a recognition by the students that technical reviews are really a form of testing that is essential during software development and maintenance. The degree of formalism, scheduling and generally positive attitude afforded to testing must exist for software technical reviews if quality products are to be produced.

i. Objectives

ii. Human-based versus machine-based

iii. Attitudes and norms

d. Reviews are a way of tracking a project

The importance of technical reviews for tracking a project must be stressed. Through identification of deliverables with well defined entry and exit criteria and successful review of these deliverables, progress on a project can be followed and managed more easily [Fagan76], [McConnell84]. In essence, review processes provide milestones with teeth. This tracking is very beneficial for both project management and customers.

i. Individual developer tracking

ii. Management tracking

iii. Customer tracking

e. Reviews provide feedback

The instructor should discuss and provide examples about the value of review processes for providing feedback about the software and its development process. Examples about how review processes can impact the existing software development such as by identifying weaknesses in the software that will require additional validation effort in the future must also be provided.

i. Product

ii. Process

f. Educational aspects of reviews

It is important in this section to stress the educational benefits of performing technical reviews [McKissick84], [McConnell84]. These benefits include a better understanding of the software by the review participants than can be obtained by reading the documentation as well as the opportunity of acquiring additional technical skills by observing the works of others [Hart82], [Peele82].

i. Project understanding

ii. Technical skills

g. Impact on morale

This section addresses the impact of technical reviews on employee morale. The students should be made aware that this impact may be either positive or negative. For some employees, such as maintenance personnel, the reviews may provide an opportunity to gain visibility of their work and, thus, will be viewed positively. For others, the reviews may be perceived as an intrusion into their workplace.

i. Positive

ii. Negative

h. Impact on maintainability

This section presents the possible impacts of review processes on software maintainability. The very nature of a review process requires the technical aspects of the software undergoing review to be understandable to the review participants. To be understandable, the software must be well documented. One of the acknowledged benefits of technical reviews is that they force developers to produce incremental documentation necessary for the review, which might not have been produced until the end of the project when schedule constraints often reduce the quality of the documentation effort.

The review process itself also improves the developer's general understanding of the whole system, which further facilitates error diagnosis during maintenance [Hart82]. Although understandability is not the only attribute of a maintainable product, the students should be made aware that review processes at least contribute in part to a more maintainable product.

i. Better documentation

ii. Standardization

iii. Readability

3. Types of Software Technical Reviews

This section identifies a variety of software technical reviews that are possible on a project depending upon the developmental model followed, the type of software product being produced and the standards which must be adhered to. Several standards that affect review processes, such as the military standards and the IEEE standard for software quality assurance plans, should be described [IEEE80], [MILS85]. Examples of typical review processes developed by organizations to satisfy these standards should also be presented [McKissick84]. Other typical types of review processes such as post mortem reviews, which are common to almost all projects, should also be discussed. The intent of this section is for the students to realize that the types of reviews that must be performed on a project are dependent upon the specific intermediate deliverables that are produced. For example, on a military contract for an embedded computer system, certain review processes are required by standards. These, however, may, not be the same type of reviews that must be performed in an expert system development effort. Examples of different models for generating and maintaining software should be provided. An emphasis on maintenance models and their associated review processes is also important in light of the large percentage of activity associated with maintenance functions. Various application areas should also be described in the context of their review processes.

a. Development life cycle models

i. Waterfall model

ii. Rapid prototyping

iii. Iterative enhancement

iv. Maintenance activity modeling

b. Current standards

i. Military standards

ii. IEEE standards

iii. NBS standards

4. Behavioral Factors

This section discusses many of the behavioral factors that must be dealt with for a successful review. The intent of this section is for the students to understand that any review process is a human activity and as such considerable attention must be spent on human interactions.

a. Motivating reviewers

The first issue that must be addressed is how to motivate reviewers. This is a complex issue that ultimately requires an organization to evolve a *culture* in which review processes are natural. Included in this culture must be recognition for good reviewers and incentives for performing this task well. Techniques for recognizing good reviewers, such as peer evaluation, and possible incentives, such as cash awards or merit increases, should be described. It is also important to conduct review processes in a consistent manner in which every developer has their work reviewed. If any personnel are immune to the review process for any reason, serious attitude problems may emerge [Hart82].

i. Review culture

ii. Incentives

iii. Management by objectives

b. Small group theory

This section addresses several key areas from *small group theory* that are relevant to reviews. Any group, including a review group, is subject to the problems of *group think*, *group deviants*, and domination of the group by a single member. Techniques for identifying these conditions and dealing with them in reviews should be described.

i. Group deviants

ii. Group think

iii. Dominance by a review participant

c. Minimizing stress

Techniques for minimizing stress for individuals whose work is being reviewed must be explored. This also requires the development of a *review culture* in which the review process is carefully defined. Important issues to address here include how review information is utilized and who has access to this information.

i. Review culture

ii. Management participation in reviews

iii. Customer participation in reviews

iv. Well-defined review processes

d. Review logistics

The section presents a description of review logistics that contribute to a successful review. These logistics include issues regarding timing of the review, location of the review and duration of the review [Fagan76]. Many of these issues are related to limitations of an individual's attention span. Other important points concern the number of reviewers and their physical arrangement. Much research has been performed suggesting that these are important variables.

i. Time of the review

ii. Location of the review

iii. Duration of the review

iv. Number of reviewers

v. Physical arrangement of group

5. Formal versus Informal Reviews

This section differentiates between formal and informal reviews. These terms are ill-defined and must be clarified in this section. Informal reviews are meant to describe the type of review that typically occurs spontaneously among peers and in which the reviewers have no responsibility and do not produce a reveiw report. Formal reviews are characterized by carefully planned meetings in which reviewers are held accountable for their participation in the review and in which a review report containing action items is generated and acted upon [Weinberg84].

In reality there exists a broad spectrum of review processes spanning from very informal peer types of reviews to extremely formal and rigorous inspections. In any software development or maintenance project there is a need for reviews that span this spectrum. As the complexity and size of a project increases, more formal reviews are necessary for tracking the project and obtaining the right participants for the review. This section also addresses differences between external and internal reviews. External reviews usually involve the customer and are more formal than internal reviews.

The intent of this section is for the students to recognize the difference between formal and informal reviews, the need for formal reviews at critical points in a project and the ability to make the distinction between what type of review is appropriate at any point in time. Global organizational issues regarding the placement and formality of review processes in an organization's development methodology should also be discussed.

a. Terminology

b. Informal reviews

c. Need for formalism

d. External/internal reviews

e. Responsibility of reviewers

f. Review reporting and follow-up

6. Review Methodologies

There are many variations to performing technical reviews. Most of these approaches involve a group meeting to assess a work product; however, variations of reviews exist that do not require a review group meeting. One variation is called a *round robin review* in which a work product is circulated among reviewers in a round robin fashion for their comments [Hart82]. Most of this section focuses on reviews that involve a group meeting. Three major approaches for performing group oriented technical reviews should be presented to the students.

It is important for the students to understand the primary differences among walkthroughs, inspections and audits as well as their respective advantages and disadvantages. The work products typically analyzed by each of these processes and the individuals who perform these processes must also be discussed. The students should also understand how to select the appropriate review methodology for a particular review.

a. Walkthroughs

The first approach that should be described are walkthroughs. Walkthroughs are well-defined by Yourdon [Yourdon78]. Walkthroughs can be viewed as *presentation reviews* in which a review participant, usually the developer of the software being reviewed, narrates a description of the software and the remainder of the review group provides their feedback throughout the presentation. These are referred to as presentation reviews because the bulk of the feedback usually only occurs for the material actually presented at the level it is presented. Thus, advance preparation on the part of reviewers is often not detectable during a walkthrough. Walkthroughs suffer from these limitations as well as the fact that they may lead to disorganized and uncontrolled reviews.

Walkthroughs may also be stressful if the developer of the software is conducting the walkthrough. This has lead to many variations such as having someone other than the developer perform the walkthrough. It is also possible to combine multiple reviews into a single review such as a combined design and code walkthrough [Hart82].

i. Presentation reviews

ii. Mechanics and variations of the process

iii. Benefits

iv. Limitations

v. Pitfalls

b. Inspections

Inspections should be presented as a more formal approach that can be viewed more as *work product reviews*. Inspections were first performed by Fagan at IBM [Fagan76]. Inspections require a high degree of preparation for the review participants, but the benefits include a more systematic review of the software and a more controlled and less stressed meeting.

There are many variations of inspections, but all include some form of a checklist or agenda that guides the preparation of review participants and serves to organize the review meeting itself. Inspections are also characterized by rigorous entry and exit requirements for the work products being inspected.

The students should be exposed to several variations of inspections such as the IBM approach [Fagan76], the Bell Labs approach [Ackerman83], and others [Peele82], [Weinberg84], [McKissick84]. The students should understand that another major difference between walkthroughs and inspections is that an inspection process involves the collection of data that can be used to feedback on the quality of the development and review processes as well as influence future validation techniques on the software itself. There are several references that provide a good explanation of the distinction between a walkthrough and an inspection [Fagan76], [Freedman82], [Quirk85].

i. Work product reviews

ii. Mechanics and variations of the process

iii. Benefits

iv. Limitations

v. Pitfalls

c. Audits

Audits should also be described as an external type of review process. Audits serve to insure that the software is properly validated and that the process is producing its intended results [Walker79].

i. Auditing work products

ii. Auditing a process

iii. Benefits

iv. Limitations

v. Pitfalls

7. Roles of Review Participants

This section defines the specific roles that must be executed by the participants of a review. These roles will vary depending upon the specific review methodology that is being followed. These roles can be viewed as being functional, which implies that it is possible in some reviews for a participant to execute more than one role. The intent of this section is for the students to understand the responsibility of each review participant before, during and after the review. The role of review participants after the review is especially important to discuss because many errors identified during the review may not be fixed correctly by the developer. This raises the issue of who should follow up on a review and whether or not another review is necessary. Several references exist for defining the various roles of review participants [Ackerman83], [Fagan76], [Hart82], [Remus83], [Peele82], [Weinberg84], [McKissick84].

a. Review leader

The review leader is the one individual responsible for the review. This role requires scheduling the review, conducting an orderly review meeting and preparing the review report. The review leader may also be responsible for ensuring that action items are properly handled after the review process. Review leaders must possess both technical and interpersonal management characteristics. The interpersonal management qualities include leadership ability, mediator skills and organizational talents. The review leader must keep the review group focused at all times and prevent the meeting from becoming a problem solving session.

i. Responsibilities

ii. Leadership / moderation skills

iii. Preparation

iv. Guidelines

b. Recorder

The recorder role in the review process guarantees that all information necessary for an accurate review report is preserved. The recorder must digest complicated discussions and capture their essence in action items. The students should be lead to understand that the role of recorder is clearly a technical function and one that cannot be performed by a secretary. This section should also discuss techniques for insuring the reliability of the information gathered by the recorder such as taping the review, having two recorders and reviewing the review minutes.

i. Responsibilities

ii. Preparation

iii. Guidelines

c. Reviewer

The reviewer role is to objectively analyze the software and to be accountable for the review. In an inspection methodology, the reviewer must spend considerable time preparing for the review. Guidelines for reviewers, such as the importance for a reviewer of keeping in mind that the software is being reviewed and not the producer of the software, should also be presented. Techniques for reviewers to pose their questions in constructive ways should also be addressed.

i. Responsibilities

ii. Preparation

iii. Guidelines for reviewers

d. Producer

The producer role varies depending upon the review methodology. In a walkthrough, the reviewer may actually lead the meeting in an organized discussion of the software. A high degree of preparation and planning is needed in a walkthrough to present material at the proper level and pace. In an inspection methodology, the producer must also be highly prepared to respond to all points brought up by the review team. The attitude of the producers is also important; it is essential that they do not take a defensive approach.

i. Responsibilities

ii. Preparation

iii. Guidelines

8. Planning for the Review Process

This section details the planning necessary for a successful review. This planning can be described at both the organizational level and the specific review level. At the organizational level, the students should be advised of the planning necessary by management to identify the appropriate number and types of reviews that are to be performed for the project. Resources must also be allocated for accomplishing these reviews. These resources are normally allocated during the creation of the Software Development or Software Quality Assurance Plans.

a. Selecting participants

At the specific review level, the students should be presented with the detailed planning issues that must be addressed for a successful review. This planning entails selecting participants and their respective roles, scheduling the review and developing a review agenda. There are many issues involved in the selection of review participants. Selecting participants is a complex task that normally is performed by management with technical input. When selecting review participants care must be exercised to insure that each aspect of the software under review can be addressed by at least some subset of the review team.

The students should also be made aware of problems that may develop if a review group becomes too large to hold a reasonable meeting. In order to minimize stress and possible conflicts in review processes, it is important to discuss the role that a possible reviewer plays in the organization. This role may be either a formal recognized role, such as manager, or an informal role, such as "spy" for management.

i. Responsibility for review participant selection

ii. Review group size

iii. Technical expertise

iv. Formal vs. informal status of reviewers in the organization

b. Scheduling the review

Scheduling issues regarding the timing of a review must also be presented to the students. These include the fact that scheduling a review should ideally take place soon after a producer has completed the software but before additional effort is expended on work dependent upon the software.

The problems of allocating sufficient time to a review process should also be explored. One of these problems stems from the difficulty in estimating the time needed to perform the review. This problem is analogous to that of estimating the time that any meeting will last. The approach that must be taken is the same as that for estimating the time to be allocated for any meeting; that is, an agenda must be formulated and time estimated for each agenda item.

Another problem to address concerning scheduling involves the duration of the review and problems that may occur if the review is too long. This requires review processes to be focused in terms of their objectives. Review participants must understand these review objectives and their implications in terms of actual review time as well as preparation time before commiting to the review.

The students should also be made aware that some organizations have collected defect data information that suggests guidelines for allocating time for a review process. This data often takes the form of code inspection rates [Buck83]. Others have collected data that suggests a certain percentage of development effort is allocated to the review process [McConnell84].

i. Management commitment

ii. Ideal review time

iii. Allocating sufficient time for the review process

c. Developing review agendas

Another objective of this section is for the students to understand the importance of developing review agendas and to be able to create and refine an agenda for a particular review. The development of a review agenda must be done prior to the review by the review leader and the producer. Although review agendas are specific to any particular product and the objective of its review, generic agendas can and should be produced for related types of products. These agendas may take the form of checklists. For example, generic code agendas and interface specification agendas can be developed. These generic agendas can become standardized as the format and content of software development work products mature.

The support materials for this module will contain sample agendas for many types of reviews.

i. Refining the scope of the review

ii. Checklists

9. Review Reports

This section describes the contents of a review report. The intent of this section is for the students to understand the type of information that is necessary to capture from a review a report. The format of reports is not important, although, numerous examples of reports should be provided and will be included in the appendix. References containing sample reports include [Ackerman83, Buck83, Fagan76, Weinberg84]. These contents can best be discussed by examining them from a management perspective, a customer perspective, a developer perspective and a SQA perspective.

a. Management perspective

From a management perspective, the review report serves as a summary of the review that highlights what was reviewed, who did the reviewing and what was their assessment. It is somewhat controversial as to whether or not management should be concerned with actual action items. Clearly, management does need an estimate of when all action items will be resolved to successfully track the project.

i. Technical review summaries

ii. Project tracking

b. Customer perspective

The customer may be interested in analyzing review reports for some of the same reasons as the manager (i.e., for tracking the project). The customer will also often be concerned with examining the quality of intermediate work products in an effort to monitor the development organization's progress.

c. Developer perspective

In the analysis of a review report from a developer perspective, the critical information is contained in the action items. These items may correspond to actual errors, possible problems, inconsistencies or other considerations that the developer must address.

i. Action items

d. SQA perspectivᵉ

The SQA perspective of a review report is twofold. First, SQA must ensure that all action items on the review report are addressed. SQA should also be concerned with analyzing the data on the review forms and classifying defects to improve the software development and review processes. Many possible classifications of defects exist and examples should be cited [Ackerman83]. There is also a variety of interpretations that can be made from accurate review data reporting that must be discussed [Buck83]. For example, a high number of specification errors may suggest a lack of rigor or time in the requirements specifications phase of the project. Information regarding the type of review, its participants, its agenda and its scheduling should also be recorded in order to facilitate improved review planning activities. In essence, the information on review reports should be utilized to evaluate both the software and its development process. This is most often performed during some sort of post mortem review of a project.

i. Data collection

ii. Action item follow-up

10. Assessing the effectiveness of technical reviews

This section presents approaches for evaluating the effectiveness of technical reviews as well as actual review data collected by organizations [Remus83], [McKissick84]. Prior to this section, the student should have been made aware of the relative merits of review processes and the mechanics for executing them. The objective of this section is to provide the students with the knowledge of how to actually evaluate particular review processes in an organization. One informal method of assessing the effectiveness of review processes is to observe their impact on the testing process. Some studies report that effective reviews reduce total test time and production failures [Peele82]. Other studies have attempted to evaluate the effectiveness of reviews using statistical techniques. This effectiveness figure is then utilized to predict the number of remaining defects in the software [Remus79]. Two quantitative metrics for accomplishing this task, error detection efficiency and error detection cost effectiveness, should also be presented.

a. Error detection efficiency

Error detection efficiency is a simple metric, which examines the ratio of the defects detected by a review process to the number of defects that were detectable. This metric should be applied over a large number of reviews in a statistical manner. The determination of the number of defects detectable by the review process can only accurately be calculated after-the-fact. Techniques for estimating this number should, however, be presented. These techniques include industry averages, reliability measures and error-seeding techniques.

i. Defect data collection

ii. Error classification schemes

b. Cost effectiveness

Error detection cost effectiveness is a more complex metric, which examines the ratio of the costs saved by the review process to the actual cost of the process. The actual cost of a review process can be measured several ways, typically using some variation of manhours. The costs saved by a review is much harder to quantify and usually requires a statistical analysis of the errors detected by the review, where these errors may have been detected if the review was not held and the cost associated with fixing the error at a later stage than if it was caught by the review. A key objective of this section is for the students to understand these types of metrics and recognize that particular review processes in an organization may not be efficient or cost effective.

i. Measuring costs saved by the review

ii. Calculating the actual cost of a review

c. Relationship of reliability assurance techniques

The relationship of various types of reliability assurance techniques, which include both testing and reviews, must also be discussed. Examples and data indicating the relative effectiveness of testing versus review processes at particular points in a project should be presented [Remus79], [Remus83]. In particular, the impact of review and/or testing processes at one point in a project with reviews and/or testing processes at later points in the project must be examined. Reliability assurance techniques must be coordinated to maximize effectiveness.

i. Reviews

ii. Testing

d. Tool support for review processes

The role of tools in terms of improving the effectiveness of review processes should also be discussed. As tools become available to perform some of the tasks previously performed by humans, the cost effectiveness of review processes increases. Examples of how this is occuring should be cited. A simple example is the utilization of a compiler to detect syntax errors in code, thus alleviating this task for the reviewers. Design and specification consistency checkers are another example of tool support for review processes.

Teaching Considerations

Recommended Support Materials

To support the instruction of this module, example review report forms, suggested review participants lists, sample review agendas, applicable checklists, and detailed information about various types of reviews should be provided. Such materials will eventually be available in the support materials package for this module.

A video tape lasting approximately 50 minutes that addresses the behavioral factors in review processes would be valuable. This tape would consist of narrated segments of reviews that illustrate both good and bad behavioral characteristics. The purpose of the video tape is to explain some of the behavioral factors in review processes in a more professional manner than a typical instructor untrained in psychology or sociology.

Exercises

Several kinds of exercises have been found to be useful and effective in teaching this material. These are described under the appropriate heading from the outline presented previously.

Review Processes in Society. Have the student think about typical non-technical reviews encountered every day in our society and address each of the applicable subtopics outlined in this section pertinent to these reviews.

Types of Software Technical Reviews. Have the student examine several different types of products, such as an embedded system and an expert system. For each of these products, identify appropriate reviews that are applicable, assuming the project is being developed with a particular life cycle model.

Roles of Review Participants. The exercises in this section involve actual reviews. Each student should have the opportunity of playing each of the review participant roles for both an inspection and a walkthrough. The students should also have the opportunity of observing reviews and reporting back problems that they thought hindered these reviews as well as actions that enhanced the reviews.

Planning for the Review Process. Various exercises can be designed to provide the students with a better understanding of selecting review participants, estimating the time for the review and developing a workable review agenda. A reasonable exercise would be to examine the Elevator Control Problem, which is one of the continuing examples under development at the SEI, and define several reviews such as the specification review, the high level design review and a test plan review. For each of these reviews, participants should be identified, a workable agenda established and a time estimate for the review derived.

Review Reports. A reasonable set of exercises for this section would be to present the students with comprehensive summary data collected from various review processes and ask them to interpret the data in terms of both the software and its development process. Another variation of this same scheme would be to ask the students how they would assess any weaknesses in an organization's development approach and evaluate the impact of changes targeted to improve these weaknesses.

Bibliography

Ackerman83

Ackerman, A. F., P. Fowler, and R. Ebenau. "Software Inspections and the Industrial Production of Software." *Software Validation, Inspection-Testing-Verification-Alternatives: Proceedings of the Symposium on Software Validation.* Amsterdam: North-Holland, Sept. 1983, 13-40.

Abstract: Software inspections were first defined by M.E. Fagan in 1976. Since that time they have been used within IBM and other organizations. This paper provides a description of software inspections as they are being utilized within Bell Laboratories and the technology transfer program that is being used for their effective implementation. It also describes the placement of software inspections within the overall development process, and discusses their use in conjunction with other verification and validation techniques.

The inspection processes at Bell Laboratories is presented. Sample reports are included along with estimates for reviewing lines of code.

Boehm76

Boehm, B. "Software Engineering." *IEEE Trans. Computers C-25,* 12 (Dec. 1976), 1226-1241.

Provides data on increasing error costs the later errors are detected and repaired in the software life cycle.

Buck83

Buck, R., and J. Dobbins. "Application of Software Inspection Methodology in Design and Code." *Software Validation, Inspection-Testing-Verification-Alternatives: Proceedings of the Symposium on Software Validation.* Amsterdam: North-Holland, Sept. 1983, 41-63.

Another IBM variation of the inspection process for design and code is detailed. Sample reports are included. A discussion of how to interpret data as the result of review processes is also included.

Deutsch88

Deutsch, M. and R. Willis. *Software Quality Engineering: A Total Technical and Management Approach.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

Table of Contents
Part I
Quality Concepts
Part II
Engineering-In Quality
Part III
Using Verification and Validation to Review-Out Defects and Test-Out Errors
Part IV
Management Aspects of Software Quality

A helpful presentation of concepts of software quality engineering. The chapters on formal technical review processes present strengths and weaknesses of alternative techniques for technical reviews by groups.

"Sample Software Quality Requirements Specification" (20 pages) in an Appendix provides an extended definition of software quality.

Fagan76

Fagan, M. "Design and Code Inspections to Reduce Errors in Program Development." *IBM Systems J. 15,* 3 (1976).

A must read classic paper that introduces the whole concept of software inspections. Sample forms, checklists and experimental data from IBM are also presented.

Freedman82

Freedman, D., and G. Weinberg. *Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products.* Boston: Little, Brown, 1982.

This text is written as a series of questions and answers. It describes the Fagan methodology and many aspects of review processes. It provides a discussion of how to review many typical documents. It is weak in its discussion of sociological factors, review reports, planning issues and assessment of reviews.

Hart82

Hart, J. "The Effectiveness of Design and Code Walkthroughs." *Proceedings of COMPSAC '82. IEEE Computer Society's Sixth International Computer Software and Applications Conference.* Silver Spring, MD: IEEE Computer Society Press, Nov. 1982, 515-522.

Many benefits of performing design and code walkthroughs are cited. Variations of reviews, including "round robin reviews," and their relative effectiveness are also noted. Actual sociological problems encountered at Sperry are also briefly mentioned.

IEEE80

IEEE Standard for Software Quality Assurance Plans. IEEE Computer Society Press, Silver Spring, MD, 1980.

The IEEE standard for Software Quality Assurance puts review processes into perspective with the entire software quality assurance process. Specific reviews are mandated by this standard.

McConnell84

McConnell, P., and W. Strigel. "Results of Modern Software Engineering Principles Applied to Small and Large Projects." *AFIPS Conference Proceedings of the 1984 National Computer Conference.* Montvale, NJ: AFIPS Press, July 1984, 273-281.

Abstract: This paper discusses the software development environment tools, techniques, and methodology as applied in two mediums to large real-time software projects. Both quantitative and qualitative measures of success obtained in these projects are discussed. The quantitative measures are statistics representing the size of produced code, the manpower over the project life cycle, and other data relevant to software engineering management. The qualitative evaluation is more concerned with results obtained from walkthroughs and various aspects of the applied methodology. Results are compared with those reported in the literature. Recommendations and suggestions for further improvements are presented.

The impact of review processes and their cost to implement on two medium to large real-time software projects are documented. The utilization of review processes to track a project is also described.

McKissick84

McKissick, J., M. Somers, and W. Marsh. "Software Design Inspection for Preliminary Design." *Proceedings COMPSAC '84. The IEEE Computer Society's Eighth International Computer Software and Applications Conference.* Silver Spring, MD: IEEE Computer Society Press, Nov. 1984, 518-519.

Abstract: The continuing need for improved computer software demands improved software development techniques. A technique for the inspection of preliminary software designs is described. Experience and results from the application of this technique are presented.

An inspection process at General Electric Company for preliminary designs is outlined including the roles of the review participants. The benefits of this process, including improved education, are also cited.

MILS85

Military Standard for Technical Reviews and Audits for Systems, Equipments, and Computer Software. United States Department of Defense, 1985. MIL-STD-1521B.

This standard defines the required reviews for military contracts. The appendices contain details about exactly what is to be covered for each of the mandated reviews as well as the role of the contractor and the contracting agency.

Peele82

Peele, R. "Code Inspections at First Union Corporation." *Proceedings of COMPSAC '82. IEEE Computer Society's Sixth International Computer Software and Applications Conference.* Silver Spring, MD: IEEE Computer Society Press, Nov. 1982, 445-446.

Abstract: At First Computer, a code inspection is conducted after the coding of a program or module is complete as indicated by a clean compilation of the program and prior to unit testing of the program. The completed program specifications and a clean compilation are the entry criteria for the inspection process. An inspection team at First Computer consists of four members; one moderator and three inspectors. The moderator is the key person in the process with the responsibility to ensure the best possible review of the program. The moderator approves the team members for the inspection and makes the necessary decisions related to scheduling and conducting the sessions. The moderator is the facilitator of the inspection meetings but is also an active participant charged with finding defects. The moderator must log all defects found during the sessions, ensure that all defects found are corrected by the author, and decide whether or not to reinspect the code.

This paper presents a variation of the Fagan inspection methodology defining the process in depth along with the roles of the review participants. The benefits of utilizing their process are also documented.

Quirk85

Quirk, W. J, ed. *Verification and Validation of Real-Time Software.* Berlin: Springer-Verlag, 1985.

This text concentrates on testing techniques for real-time software. The utilization of review processes is also described. The emphasis of these review processes is, however, not unique to real-time software and very little insight into reviewing real-time systems as opposed to other types of systems can be obtained from this text.

Remus79

Remus, H., and S. Zilles. "Prediction and Management of Program Quality." *IEEE Proceedings of the Fourth International Conference on Software Engineering*. Silver Spring, MD: IEEE Computer Society Press, Sept. 1979, 341-350.

Abstract: Techniques such as design reviews, code inspections, and system testing are commonly being used to remove defects from programs as early as possible in the development process. The objective of the authors is to demonstrate that predictors can be devised which tell us how well defects are being removed during the defect removal process.

The paper presents statistical techniques for estimating the number of errors remaining in a product based on data collected from reviews. Approaches for evaluating reviews and the relationship of various reviews to each other and to testing are also described.

Remus83

Remus, H. "Integrated Software Validation in the View of Inspections/Reviews." *Software Validation, Inspection-Testing-Verification-Alternatives: Proceedings of the Symposium on Software Validation*. Amsterdam: North-Holland, Sept. 1983, 57-63.

Abstract: The software development process is looked at as to the specific contribution of inspections/reviews to the discovery of wrong design directions or implementations. The benefits are evaluated under the aspects of quality/productivity improvement and/or cost savings.

The relationship of review processes to testing in an IBM environment are explored. A variation of the roles of the review participants is presented as well. The utilization of defect data to the discovery of wrong design directions and implementations is also described.

Walker79

Walker, M. "Auditing Software Development Projects: A Control Mechanism for the Digital Systems Development Methodology." *Proceedings, COMPCON Spring*. Silver Spring, MD: IEEE Computer Society Press, 1979, 310-314.

Software audits and their function in a development organization are defined. Auditing techniques are presented as well as experiences from the Computer Science Corporation.

Weinberg84

Weinberg, G., and D. Freedman. "Reviews, Walkthroughs, and Inspections." *IEEE Trans. Software Eng. SE-10*, 1 (Jan. 1984), 68-72.

Abstract: Formal technical reviews supply the quality measurement to the "cost effectiveness" equation in a project management system. There are several unique formal technical review procedures, each applicable to particular types of technical material and to the particular mix of the Review Committee. All formal technical reviews produce reports on the overall quality for project management, and specific technical information for the producers. These reports also serve as an historic account of the systems development process. Historic origins and future trends of formal and informal technical reviews are discussed.

An overview paper describing the distinction between walkthroughs and inspections. The difference between formal and informal reviews is also clarified. The paper also contains sample review reports and how these reports can be used.

Yourdon78

Yourdon, E. *Structured Walkthroughs*. New York: Yourdon, Inc., 1978.

A detailed discussion of the walkthrough process. The benefits of walkthroughs as well as the mechanics of the process are presented. Psychological issues for walkthroughs are also noted.

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

EPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
NONE

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
SEI-CM-3 1.5

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
SOFTWARE ENGINEERING INST.

6b. OFFICE SYMBOL *(If applicable)*
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI JOINT PROGRAM OFFICE

6c. ADDRESS *(City, State and ZIP Code)*
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA 15213

7b. ADDRESS *(City, State and ZIP Code)*
ESD/AVS
HANSCOM AIR FORCE BASE, MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
SEI JOINT PROGRAM OFFICE

8b. OFFICE SYMBOL *(If applicable)*
ESD/ AVS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F1962890C0003

8c. ADDRESS *(City, State and ZIP Code)*
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
63752F	N/A	N/A	N/A

11. TITLE *(Include Security Classification)*
The Software Technical Review Process

RSONAL AUTHOR(S)
James S. Collofello, Arizona State University

13a. TYPE OF REPORT
FINAL

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
June 1988

15. PAGE COUNT
14

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
technical review   walkthrough
inspection

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This module consists of a comprehensive examination of the technical review process in the software development and maintenance life cycle. Formal review methodologies are analyzed in detail from the perspective of the review participants, project management and software quality assurance. Sample review agendas are also presented for common types of reviews. The objective of the module is to provide the student with the information necessary to plan and execute highly efficient and cost effective technical reviews.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS ☒

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED, UNLIMITED DISTRIBUTION

22a. NAME OF RESPONSIBLE INDIVIDUAL
JOHN S. HERMAN, Capt, USAF

22b. TELEPHONE NUMBER *(Include Area Code)*
412 268-7630

22c. OFFICE SYMBOL
ESD/AVS
(SEI JPO)

The Software Engineering Institute (SEI) is a federally funded research and development center, operated by Carnegie Mellon University under contract with the United States Department of Defense.

The SEI Software Engineering Curriculum Project is developing a wide range of materials to support software engineering education. A *curriculum module* (CM) identifies and outlines the content of a specific topic area, and is intended to be used by an instructor in designing a course. A *support materials* package (SM) contains materials related to a module that may be helpful in teaching a course. An *educational materials* package (EM) contains other materials not necessarily related to a curriculum module. Other publications include software engineering curriculum recommendations and course designs.

SEI educational materials are being made available to educators throughout the academic, industrial, and government communities. The use of these materials in a course does not in any way constitute an endorsement of the course by the SEI, by Carnegie Mellon University, or by the United States government.

Permission to make copies or derivative works of SEI curriculum modules, support materials, and educational materials is granted, without fee, provided that the copies and derivative works are not made or distributed for direct commercial advantage, and that all copies and derivative works cite the original document by name, author's name, and document number and give notice that the copying is by permission of Carnegie Mellon University.

Comments on SEI educational materials and requests for additional information should be addressed to the Software Engineering Curriculum Project, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Electronic mail can be sent to education@sei.cmu.edu on the Internet.

Curriculum Modules (* Support Materials available)

CM-1	[superseded by CM-19]
CM-2	Introduction to Software Design
CM-3	The Software Technical Review Process*
CM-4	Software Configuration Management*
CM-5	Information Protection
CM-6	Software Safety
CM-7	Assurance of Software Quality
CM-8	Formal Specification of Software*
CM-9	Unit Testing and Analysis
CM-10	Models of Software Evolution: Life Cycle and Process
CM-11	Software Specifications: A Framework
CM-12	Software Metrics
CM-13	Introduction to Software Verification and Validation
CM-14	Intellectual Property Protection for Software
CM-15	Software Development and Licensing Contracts
CM-16	Software Development Using VDM
CM-17	User Interface Development*
CM-18	[superseded by CM-23]
CM-19	Software Requirements
CM-20	Formal Verification of Programs
CM-21	Software Project Management
CM-22	Software Design Methods for Real-Time Systems*
CM-23	Technical Writing for Software Engineers
CM-24	Concepts of Concurrent Programming
CM-25	Language and System Support for Concurrent Programming*
CM-26	Understanding Program Dependencies

Educational Materials

EM-1	Software Maintenance Exercises for a Software Engineering Project Course
EM-2	APSE Interactive Monitor: An Artifact for Software Engineering Education
EM-3	Reading Computer Programs: Instructor's Guide and Exercises